

TIE-02306 (ItSE)

Introduction to Software Engineering

5 credit units

13-recap-ItSE2019-v2

Course contents (plan)

1. Course basics, intro
2. Sw Eng in general, overview
3. Requirements
4. Different software systems
5. Basic UML Diagrams ("Class", Use Case, Navigation)
6. Life Cycle models
7. UML diagrams, in more detail
8. Quality and Testing
9. Project work
10. Project management
11. Open source, APIs, IPR
12. Embedded systems
- 13. Recap**

13. Recap, summary, resume

What you should remember from this course...

- life cycle phases in general (analysis, requirements, design, implementation, testing, deployment, maintenance), and supporting tasks: hw, documentation, training
- same phases on all SDLC models; requirements, implementation, testing, deployment
-
- three classes of requirements; functional, non-functional, restrictions
-
- common problems in sw dev projects (not much technical, but "soft" issues)
- common success factors in sw dev projects
-
- **customer role is important, has also some responsibilities !**
-

Current at course (w 49)

- **WEs ended, nothing this week**
- **continue updating (finalise) your Trello (kanban) board (process)**
- **group-to-group feedback at PRP was open up to 01.12.2019**
- **final versions of documentation deadline is Sun 08.12.2019**
- **peer feedback inside own group, and self assesment, DL = 15.12.2019**
- **EXAM 3/3 at weeks 47-49; lectures 7-10, 12 questions (right/wrong) in 6 sections + volunteer feedback.**
(first tryouts: 18..38 min, 12/24 points)

Backlog items with deadline

- 09.09.2019 at 23:59 Group forming (Moodle)
- 15.09.2019 at 23:59 Trello creation (Trello)
- 13.10.2019 at 23:59 Phase 1 documentation (Moodle)
- 13.10.2019 at 23:59 Phase 1 presentation slides (PRP-tool)
- **Week 43** **Phase 1 presentations (Physical realm)**
- 03.11.2019 at 23:59 Phase 1 peer feedback (PRP-tool)
- 17.11.2019 at 23:59 Phase 2 documentation (Moodle)
- 17.11.2019 at 23:59 Phase 2 presentation slides (PRP-tool)
- **Week 47** **Phase 2 presentation (Physical realm)**
- 01.12.2019 at 23:59 Phase 2 peer feedback (PRP-tool)
- 08.12.2019 at 23:59 Final delivery of project documentation (Moodle)
- 15.12.2019 at 23:59 Final peer feedback and self assessments (PRP-tool).
- *and then KAIKU feedback to course... (09.12.2019 – 22.01.2020)*

03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

15



Weekly exercise attendees

	w36 WE1	w37 WE2	w38 WE3	w39 WE4	w40 WE5	w41 WE6	w44 WE7	w45 WE8	w46 WE9	w48 WE10
WED	0	14	9	5	8	9	9	7	7	6
THU	21	13	14	17	16	13	6	17	12	6

We will continue two Weekly Exercise groups, as long as the number of attendees are reasonable.

About TIE-02306, 2019 (1st run)

- EXAM 1/3 , 4..35 min time, 3,5..6 points
- EXAM 2/3 , 53..115 min time, 2,5..6 points
- EXAM 3/3 still open...
-
- A lot of material at lecture slides, extras for those who will continue major at ICT
- allright, "core business" should be better visible from "extras"
-
-
-

recommended TUNI courses

- KIE-07001 Puheviestintä ja neuvottelutaito (communication and negotiation skills)
- TIE-21201 Testaus (testing)
- TIE-20200 Ohjelmistojen suunnittelu (software design)
- TIE-21107 Sw Eng Methodologies
- TIE-22201 Tietokantojen suunnittelu (database design)
- TIE-30151 Kyberturvallisuus 1 (cyber security)
- TIE-30302 Kyberturvallisuus 2 (cyber security)
- TIE-52107 IoT (new at Spring 2020)
- TIE-23516 Basic web applications
- TIE-23536 Cloud applications
- TIE-23546 Cloud platforms
-

Remember also UN recommendations at all work



Mathematics and physics courses teach logical engineering thinking

Three skills needed most at work:

- attitude/will to study/learn more
- problem solving skills
- teamworking skills.

3 tärkeintä taitoa

Asenne oman osaamisen kehittämiseen:

- Ensiarvoisen tärkeää
- Usein korostettiin, että ratkaisevaa ei ole se mitä jo osaa vaan se mitä on halukas oppimaan

Ongelmanratkaisutaidot:

- DI/Arkkitehdin työ on isolta osalta juuri täitä
- Mainittiin myös, että yksilön ongelmanratkaisua tärkeämpää on osata ratkaista ongelmia ryhmässä

Ryhmätyötaidot:

- Työelämässä paljon asioita tehdään yhdessä, harvoin yksin
- Korostuu vielä enemmän tulevaisuudessa, ja tämän taidon opettelua enemmän osaksi opintoja

hints for studies

During your studies, **it is recommended to do groupwork**, i.e. think and solve problems with fellow students.

Our students are encouraged to solve problems themselves, but if you have been trying to solve some problem for 1..2 days, feel free to ask help from your friends.

You may "bang your head" to the wall for 1..2 days, but after that you should ask help from others.

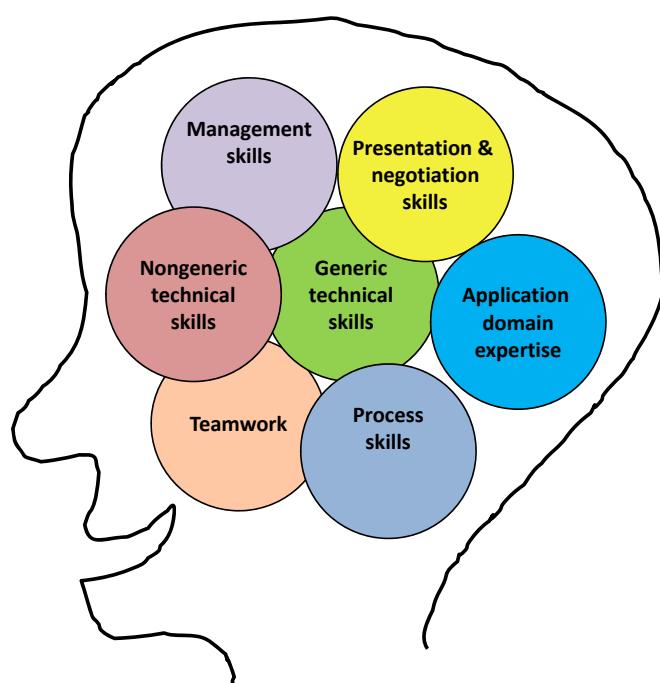
So, "never give up" can be formed to "do not give up easily".

But naturally you should not copy others' work !

1st such case = cancellation of credits, 2nd case = discharging from university.

Areas of expertise

A good software engineer (developer or customer/client) has many skills, not just technical expertise.



ISO/IEC/IEEE 15288:2015 Systems and software engineering – system life cycle processes

TIE-02306 course
is just a small part
at the beginning of
the sw sys life
cycle.



Figure 4 — System life cycle processes

03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

23

requirements, three main cases

Functional requirements , e.g.

- user's functionalities
- inputs and outputs
- browser Front and Back buttons are disabled

Non-functional requirements , e.g.

- GUI is as customer's guidebook XYZ-2019-A, coding conventions
- localisation, internationalisation = I(18)N
- safety and security

Restrictions and limitations , e.g.

- compatible with customer's old system (DB)
- supported operating systems and browsers
- project schedule and budget.

In some cases a requirement may be considered as non-functional or restriction.

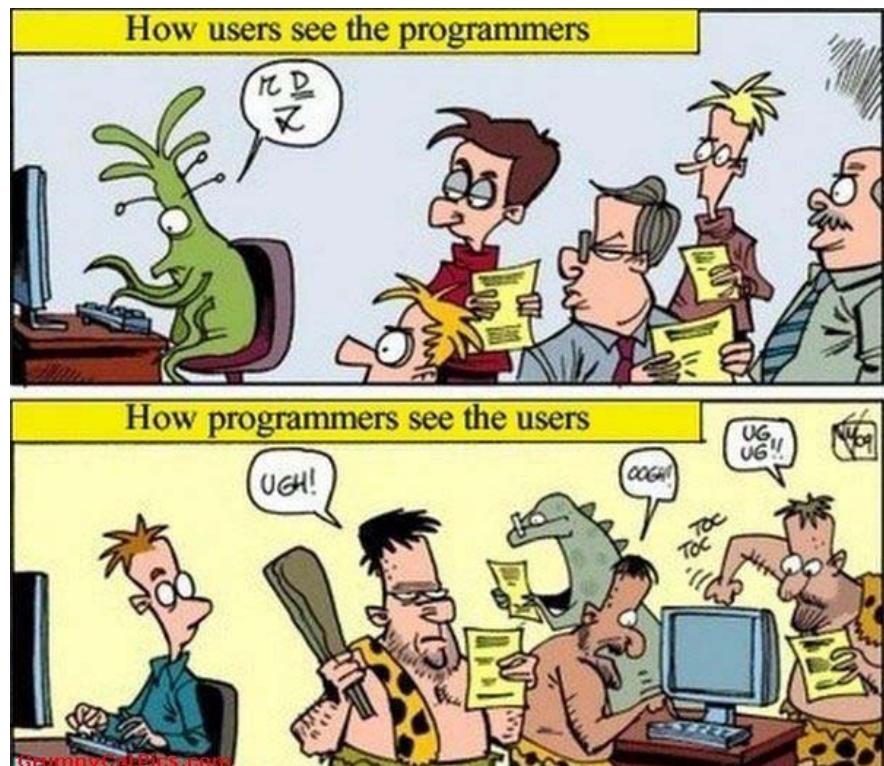


requirements

- if there would be mind-reading, not much meetings nor documentation would be needed nor required
- developer: do not "think" you know what customer or end-user wants, ask them
- customer have better be motivated and active on requirements, developers do not know everything
- it is important to list all needed requirements, in priority order
- sometimes it would be good to list things which the system does not do
-
-

**By the way,
remember
WHY you do
the software,
and to WHOM.**

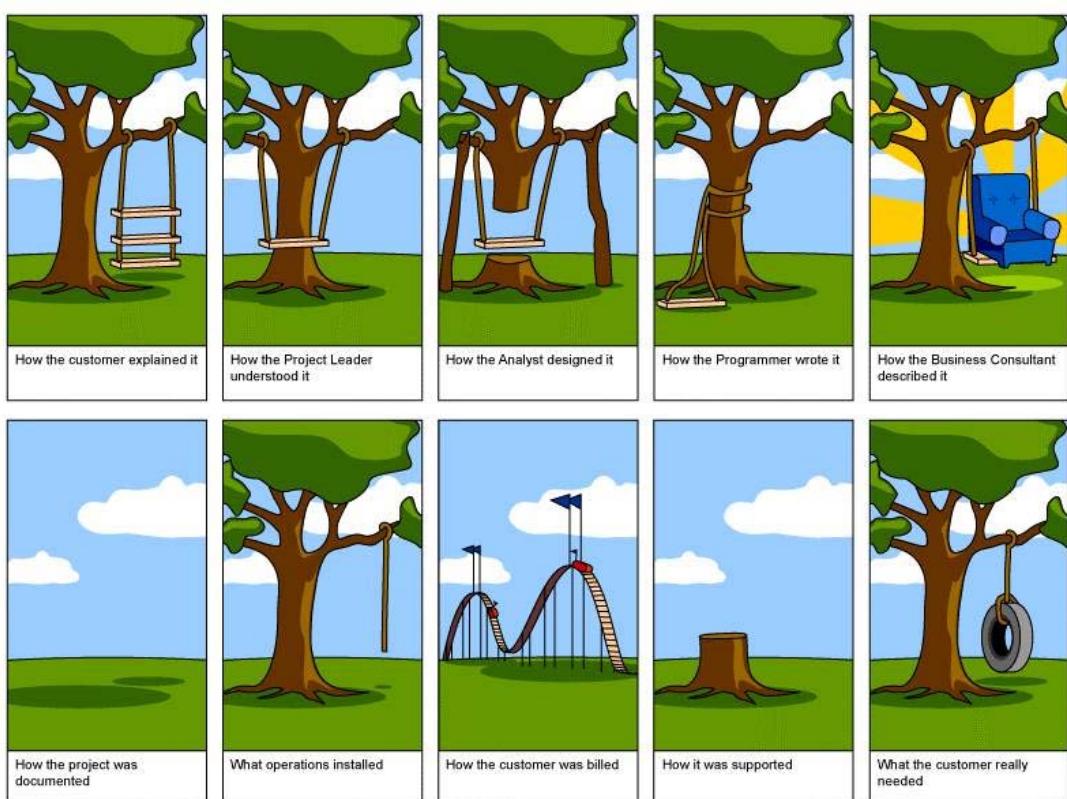
**"User before
technology...?"**



03.12.2019 15:59

TUNI TIE-02306

27



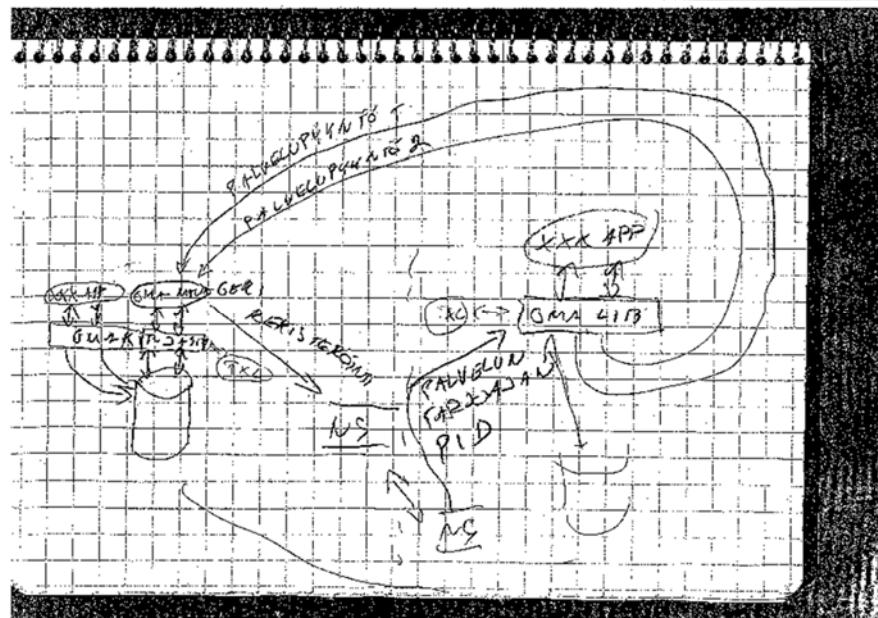
03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

28

A first developer-customer meeting memo

By the way, if you give this memo to a colleague who was not present at the meeting, would it be easy for him/her to understand what the customer's wants ?



Usually this is the start, developer continues writing preliminary requirements...

03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

29

2.2 Software Architecture Description

Different stakeholders have different views, and they do not necessarily understand each other's point of views.

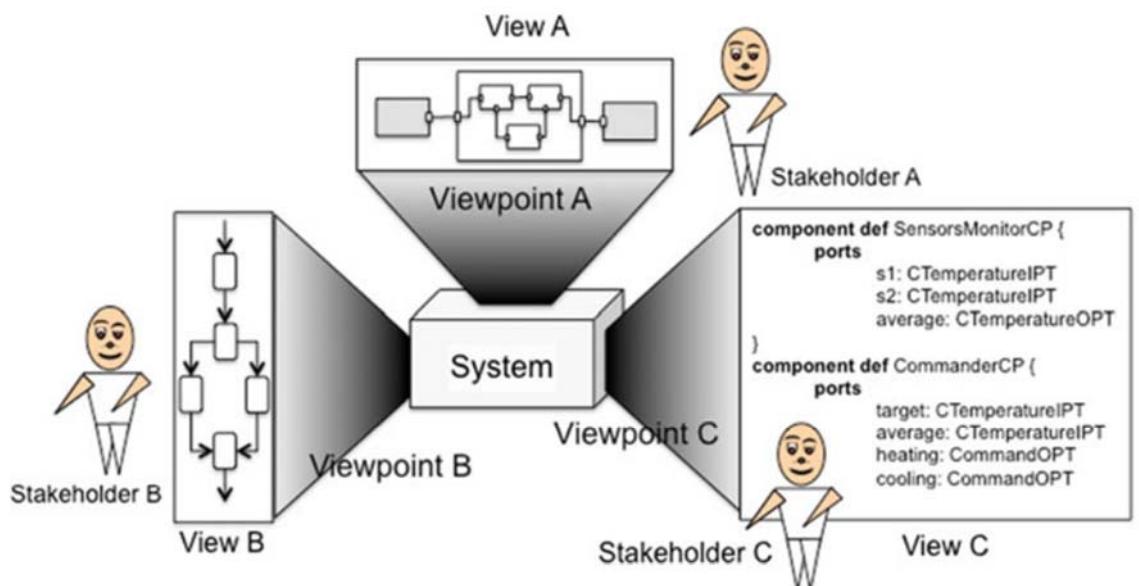
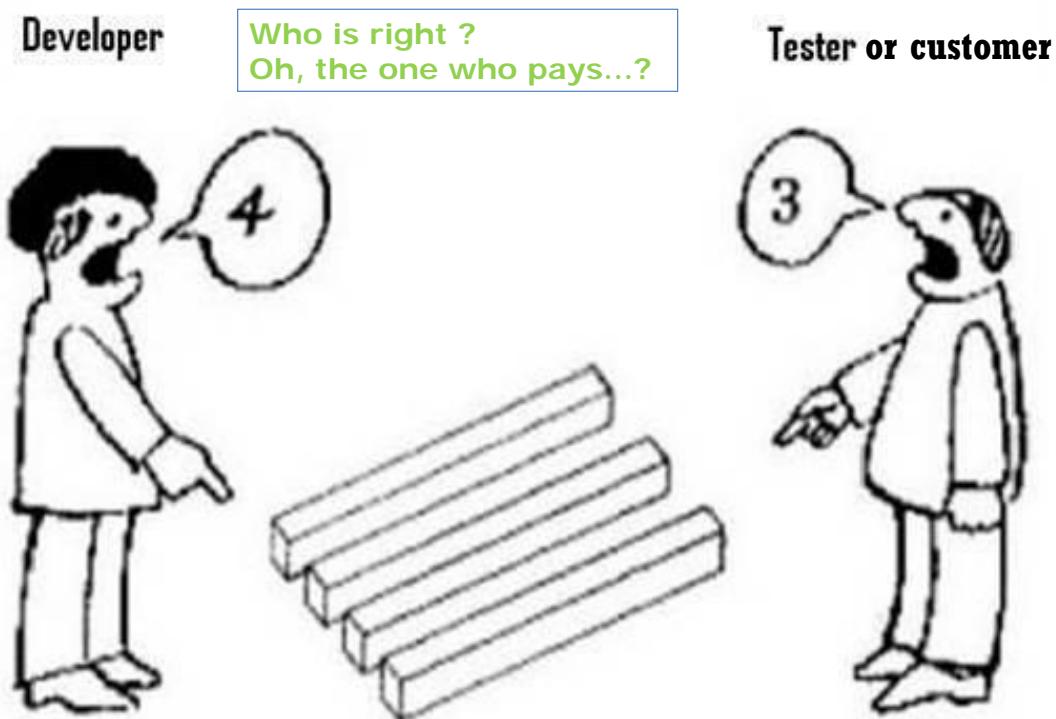
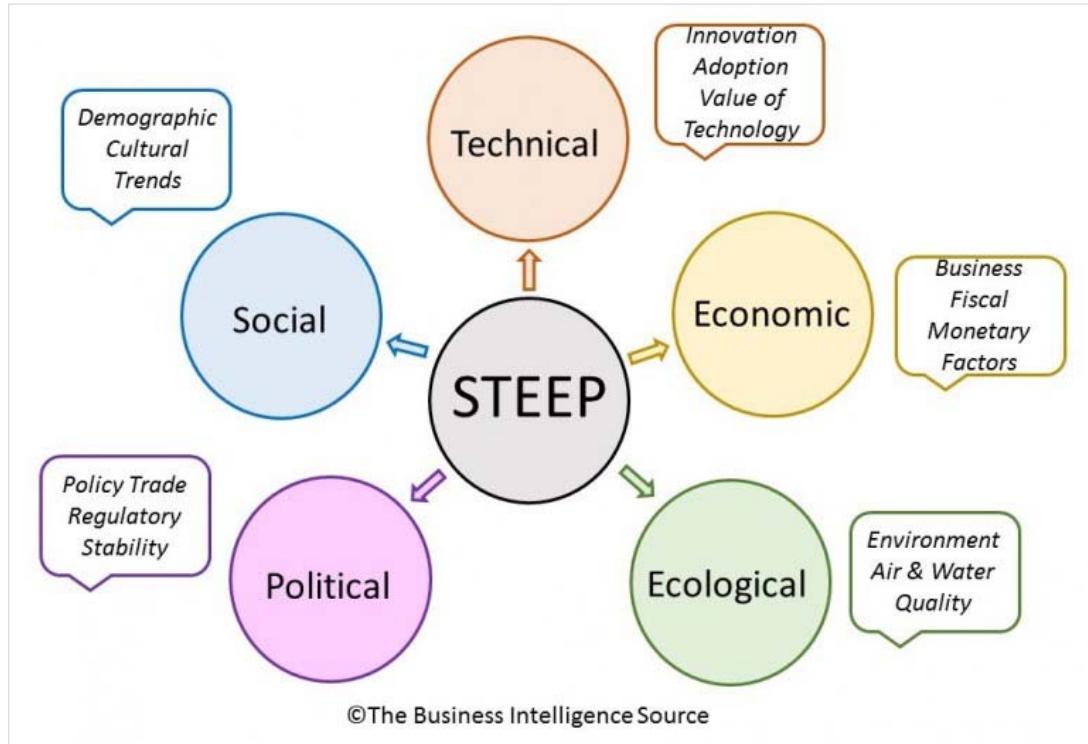
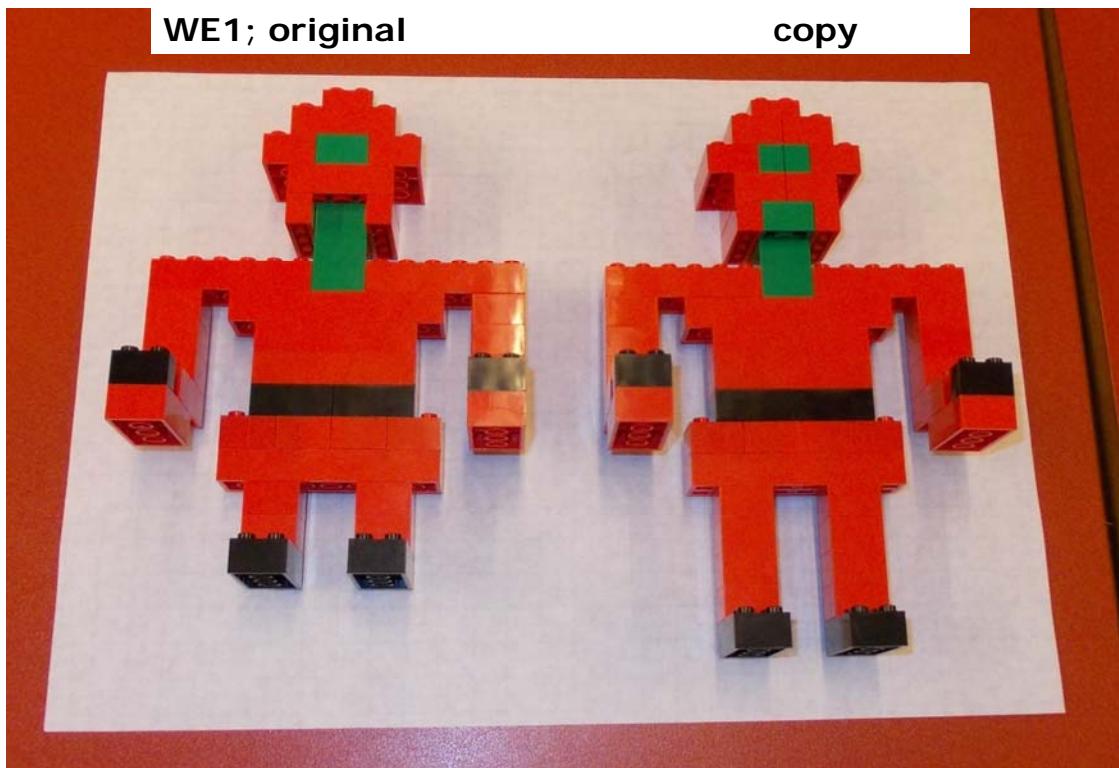


Fig. 2.5 Viewpoints and views

[Software Architecture in Action, 2016]

Stakeholder analysis,
different
viewpoints
what to
take into
account.

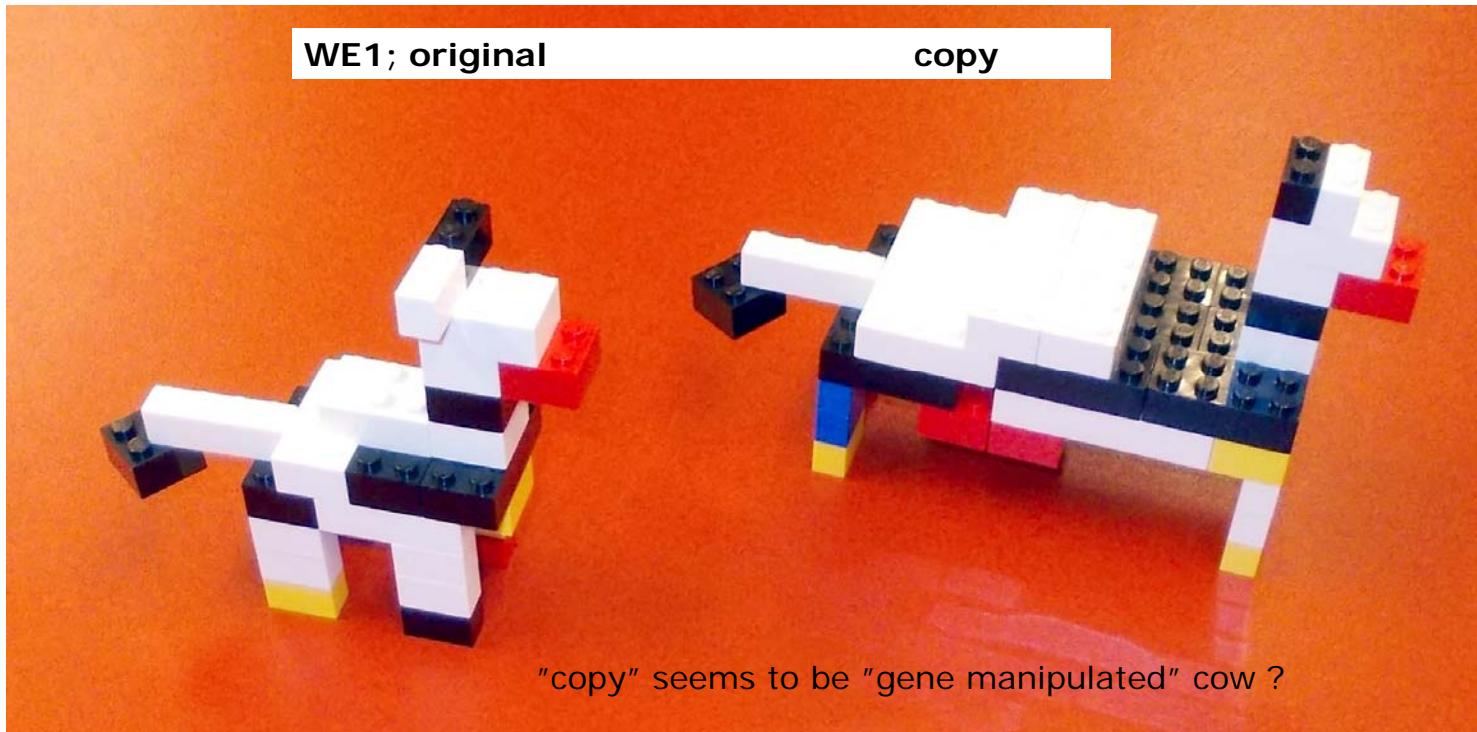




03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

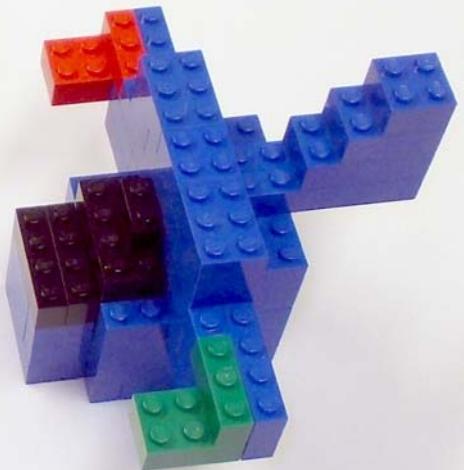
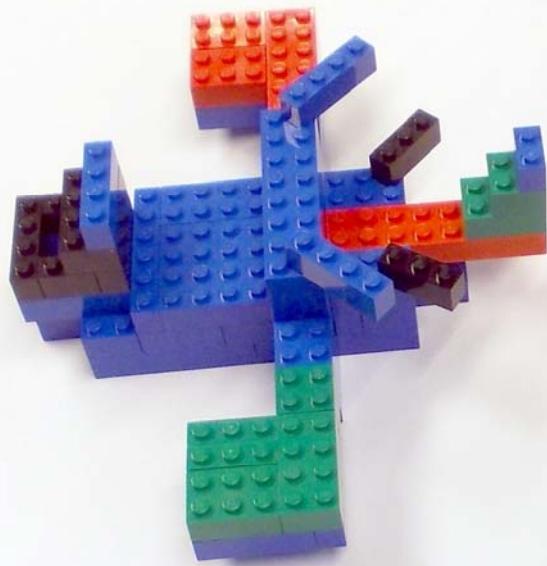
33



"copy" seems to be "gene manipulated" cow ?

WE1; original

copy

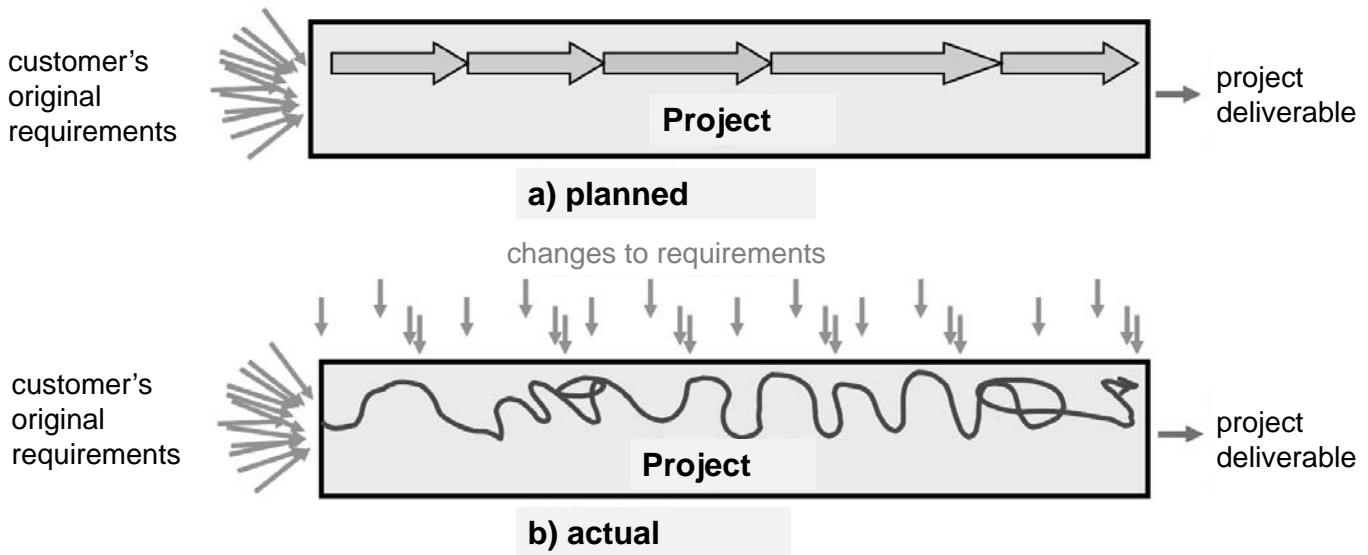


"copy" seems to be a "shortened" mini vehicle

project management

- at least 50 % of project management tasks and responsibilities are independent from the application/business area
- "project management is risk management"
- "project management is change management"
- typical risk source classes
 - project group itself
 - customer/client
 - third party / environment
 - technology.

SW project is reacting to changes and (hopefully small) surprises



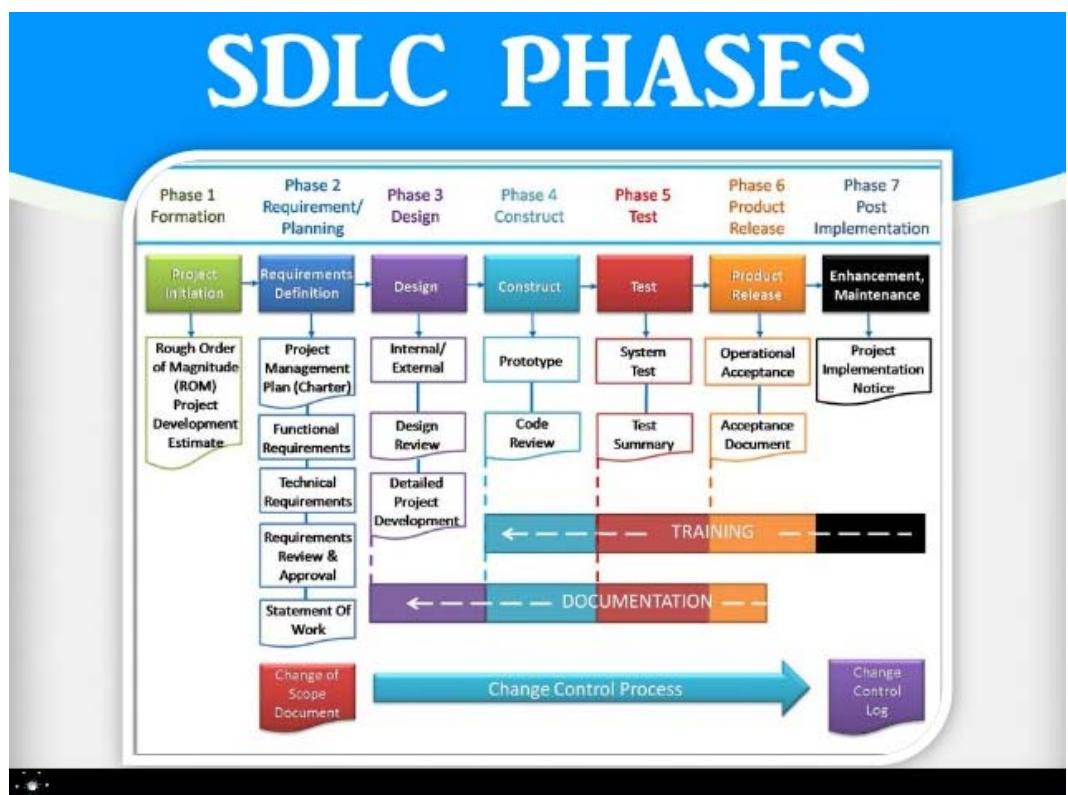
(compare that to your planned study path...?)

03.12.2019

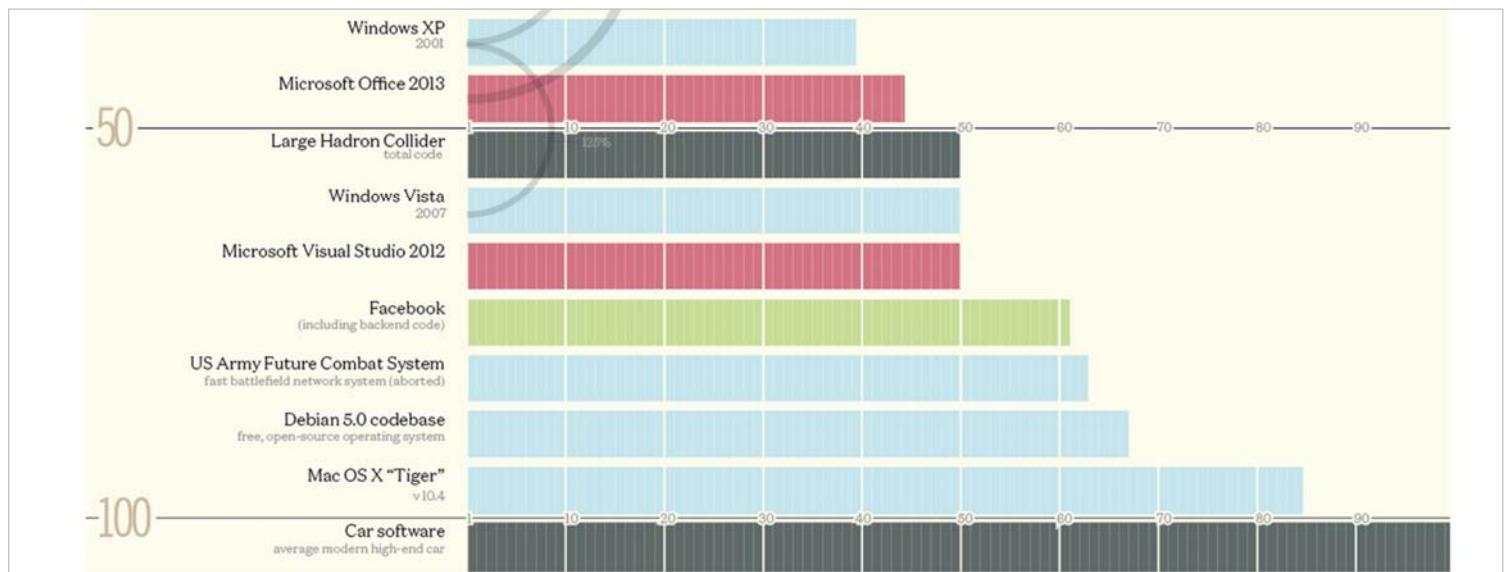
TUNI * TIE-02306 Introduction to Sw Eng

37

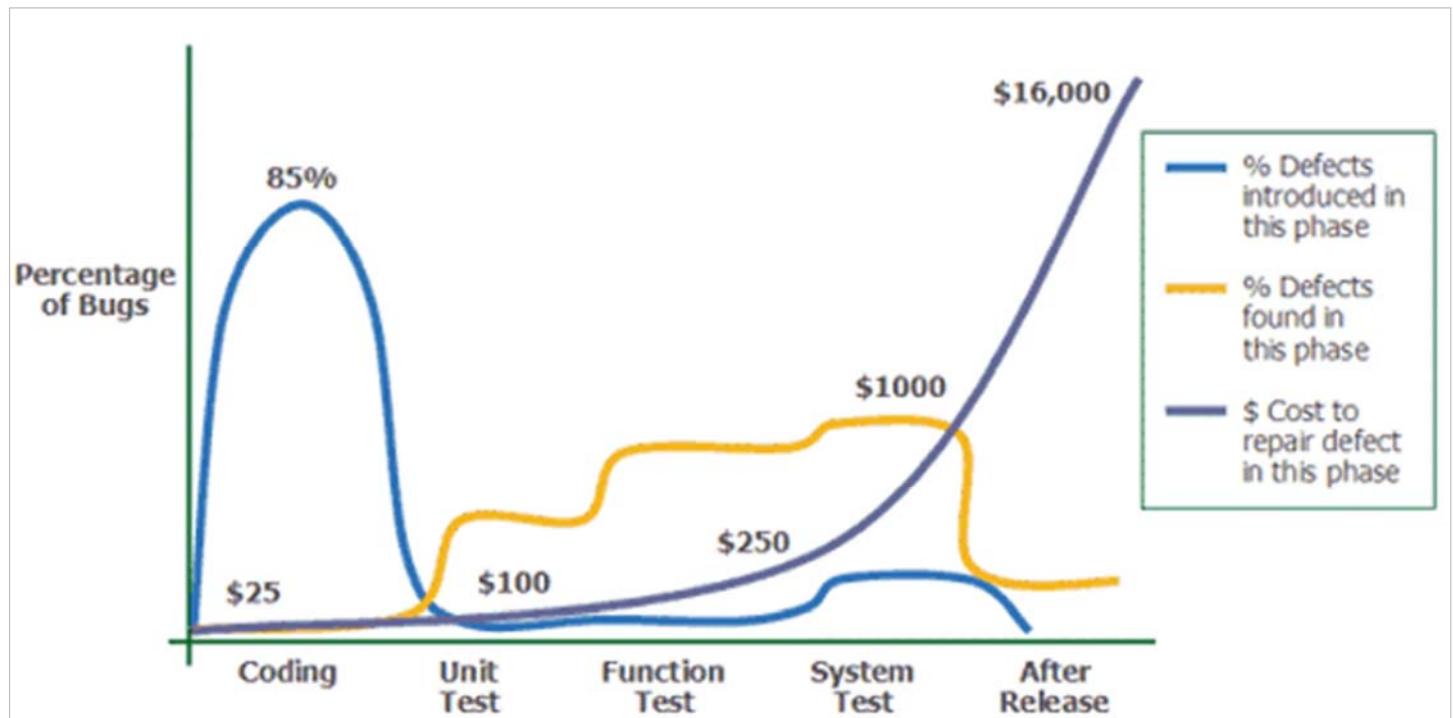
Same basic phases
 • requirements
 • implementation
 • testing
 • deployment
 exist in every life cycle model.



Some lines of code (LOC) amounts

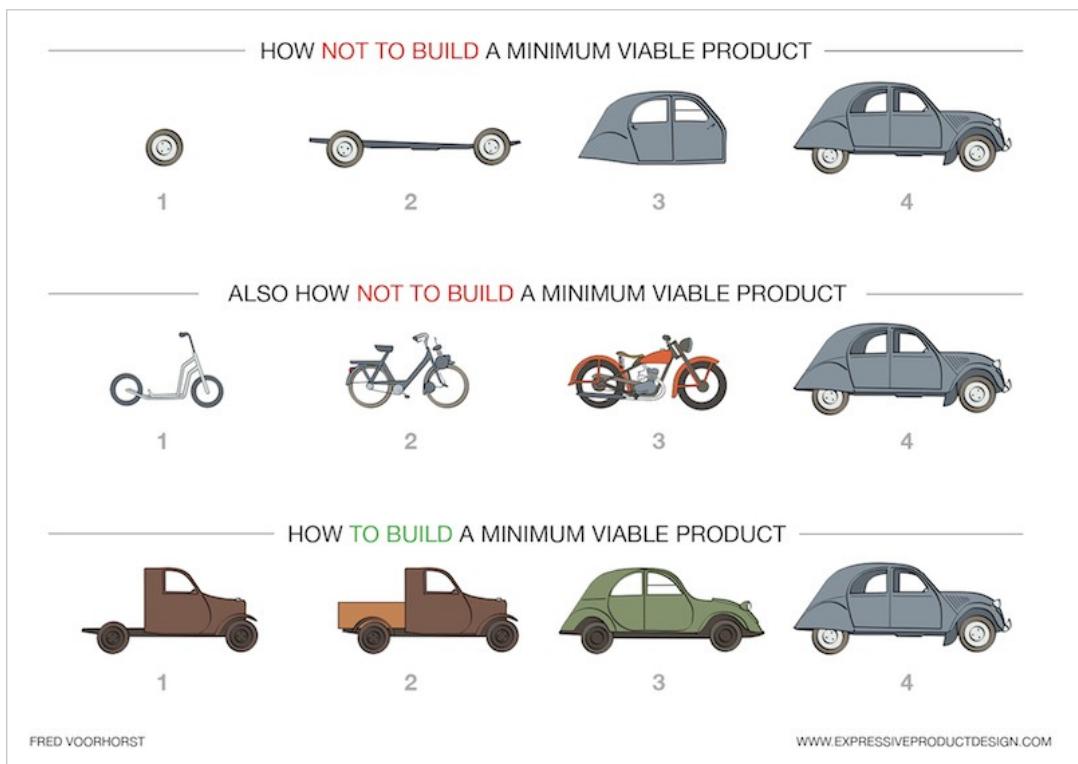


The later you find bugs, the more costly it is.

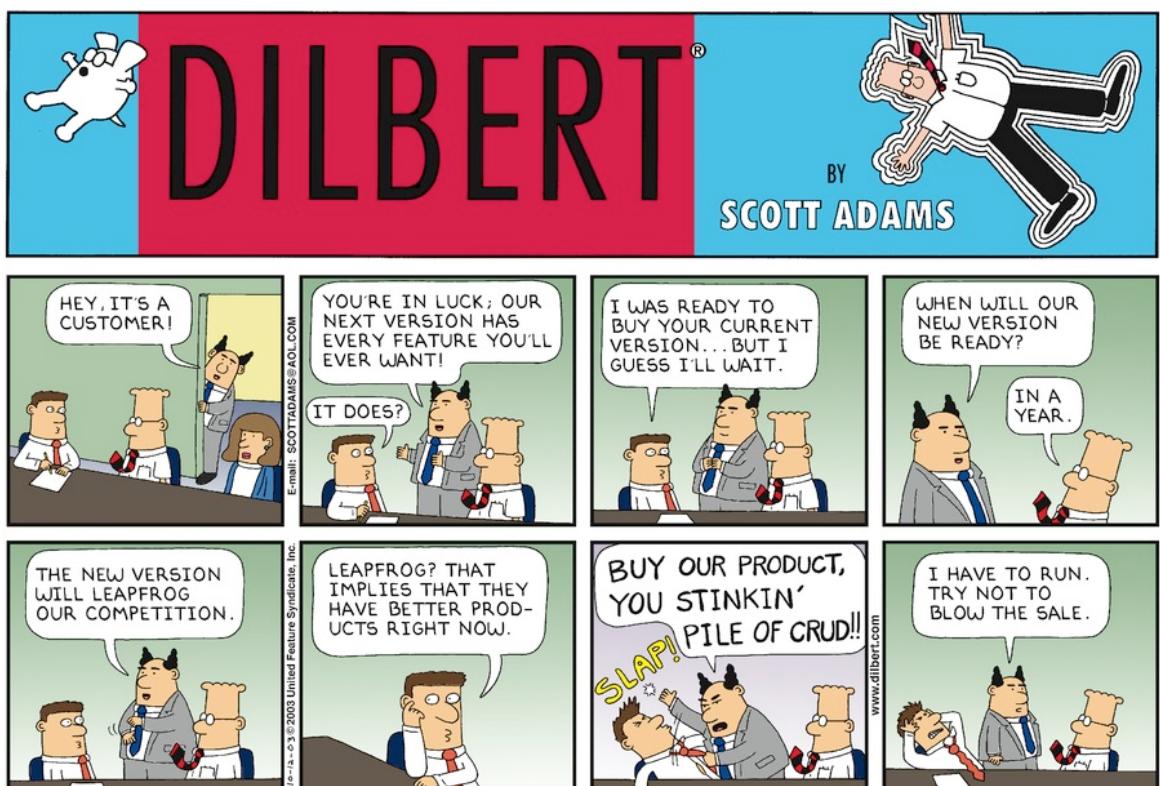


Minimum Viable Product (MVP) is NOT the "quickest hack" you dare to send to customer. This is a common misunderstanding.

MVP is the quickest made product version which brings profit/value to vendor and customer.



Luckily you need not to think about sales engineer job at this course.



What to remember from Intro to Sw Eng, 1

must know

- requirements definition process
- agile process and Scrum method
- understand the nature of software projects (e.g. difficult to estimate, humans matter)
- all in all; how to participate successfully in a software development process (either in a developer or customer role)

What to remember from Intro to Sw Eng, 2

should know

- UML diagrams (read/write)
- customer is an important stakeholder
- quality, cyber security and information security (e.g. GDPR) are with from beginning
- sw eng project problems (there are many checklists)

What to remember from Intro to Sw Eng, 3

nice to know

- all kind of techniques, methods and methodologies
- Big Data, IoT and AI are future trends (keep an eye on those)
- cyber security issues are already part of today's sw development

Implement what is at the specification, do not make any "extras" even if such would be easy



future of software development

In Finland, many more software people are needed for e.g. health care systems, both legacy and new development. Government has also many looong ICT projects going on.

In general, new technologies as IoT and AI make sure that more and more cyber security people are needed. Perhaps also software lawyers ?

IoT is currently a "Wild West", as huge amount of devices are put to market, with more or less quality embedded. Security is left to users (consumers).

Many new tools are developed every year, but how is the productivity of developers if tools are changindevery year ? How long does it take to be "productive" with a new tool ?

USER FRIENDLY by J.D. "Illiad" Frazer



diagrams

- at every software development project, there are some diagrams
- UML (Unified Modeling Language) diagrams are most common today
- at requirements gathering, you surely have use for
 - Context / Entity diagram ([FI: tietoyhteyskaavio, käsitekaavio](#))
 - Use Case diagram and User stories ([FI: käyttötapauskaavio](#))
- navigation diagrams help on GUI planning and design
- state (transition) diagrams, of several style, help on embedded and real-time sw

customer's role on projects

- customer should be committed, not just involved to the sw dev project
- customer should be motivated



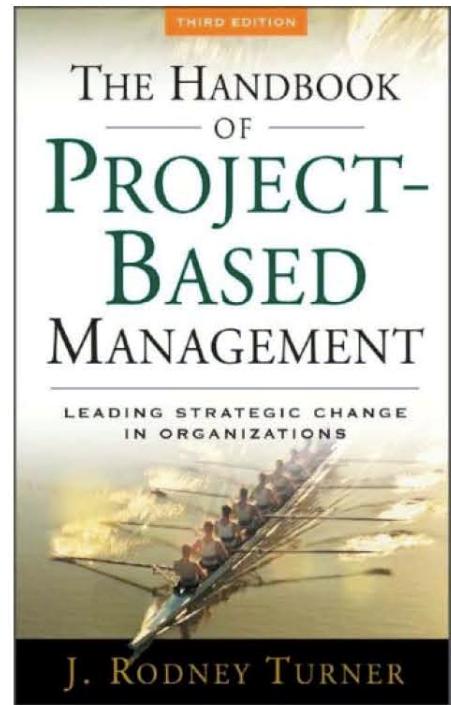
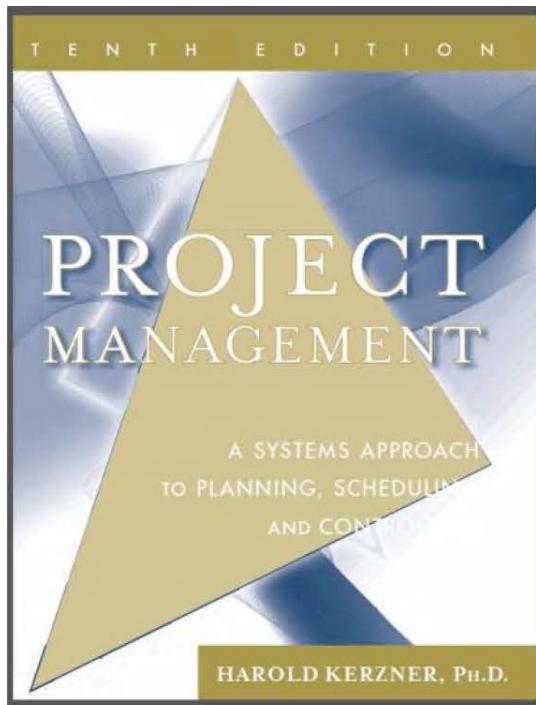
project work

- **projects are teamwork** (notice the difference between group and team)
- at the start, you have better form your group to be a TEAM
- agree communication channels and procedures
- agree tools (who is "guru" and can help others with the tool X)
- face to face (F-2-F) meetings (around the same table) are much better than remote messaging
- working together happenings, "coding nights" or "jams", are strongly recommended
- feel free to ask help from co-workers and organisation experts
-

project management

The famous general Project Management books.

At least half of all project management actions are application area independent.

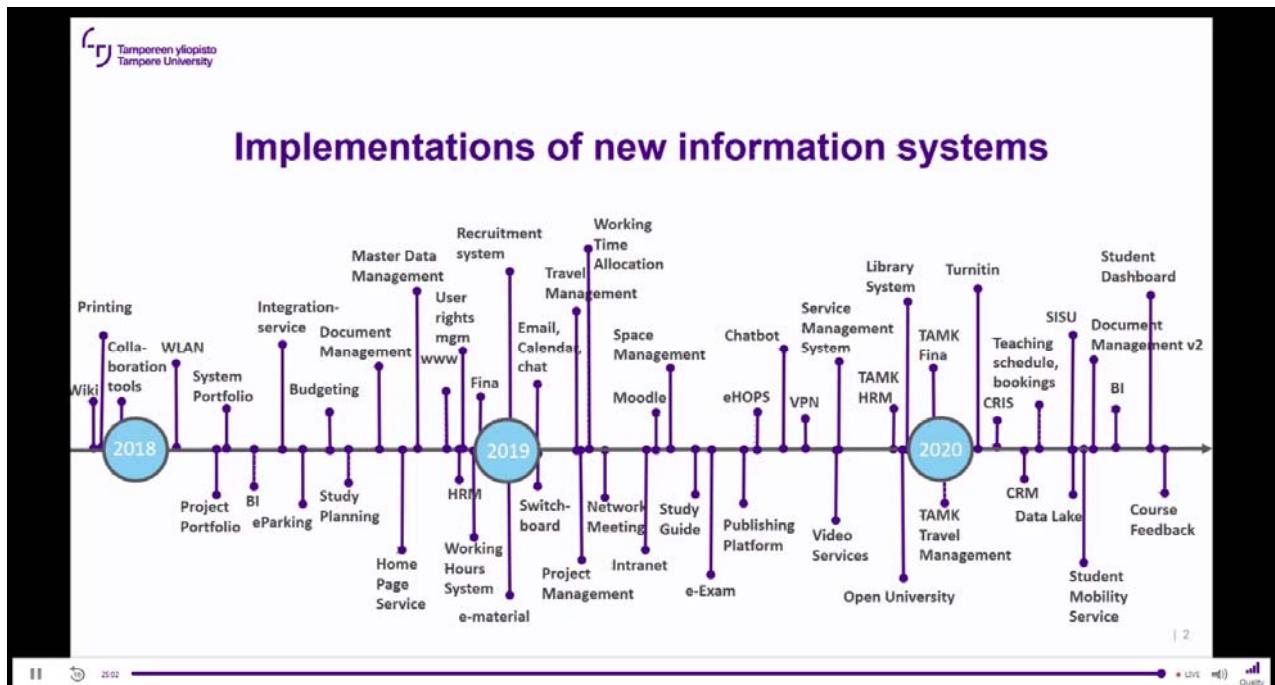


03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

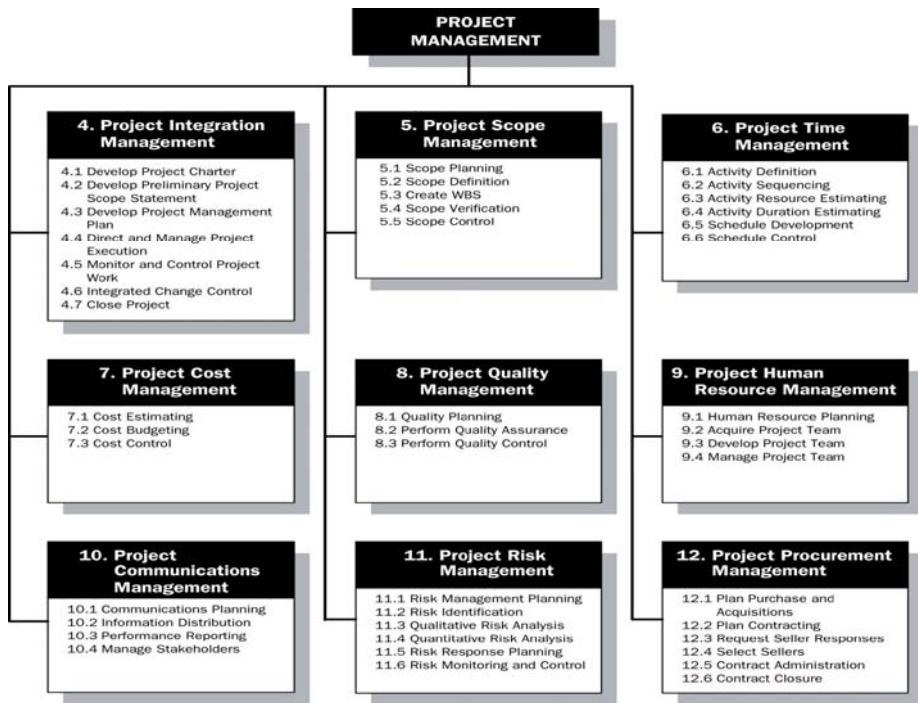
53

Big systems should be taken into use in small steps



Project management knowledge areas and processes

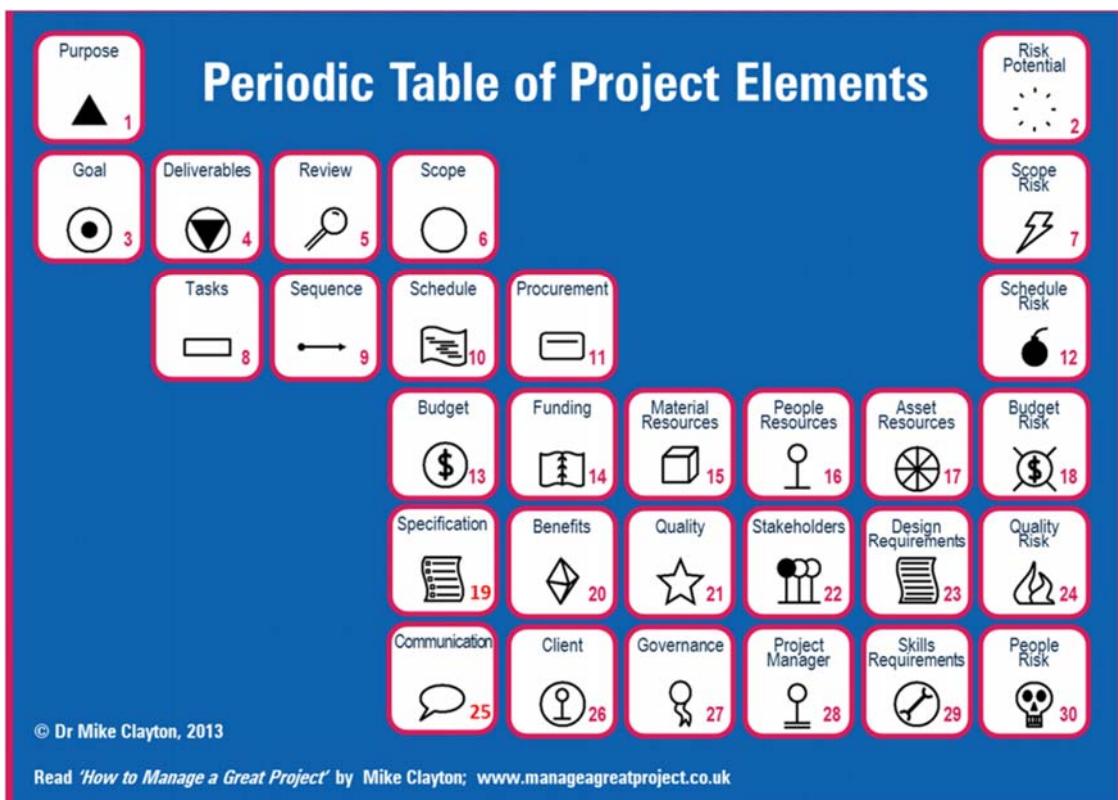
[Guide to PMBoK,
3rd ed, 2004]



03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

55



03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

56

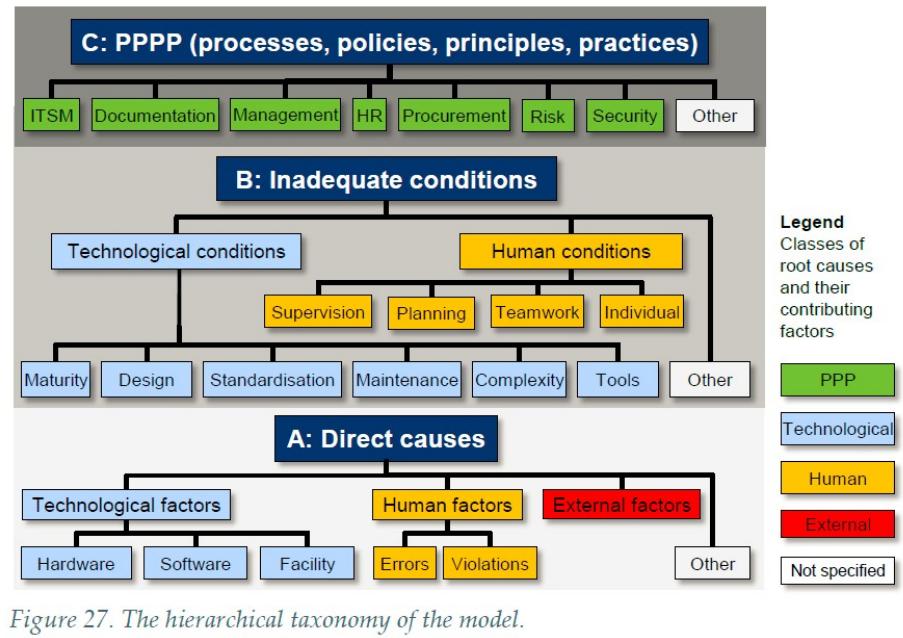


Figure 27. The hierarchical taxonomy of the model.

03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

57

Consider
those also
to your
exercise
work.



03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

58

"motivated employer makes good work"



© 2012. Intégral Leadership Institute

03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

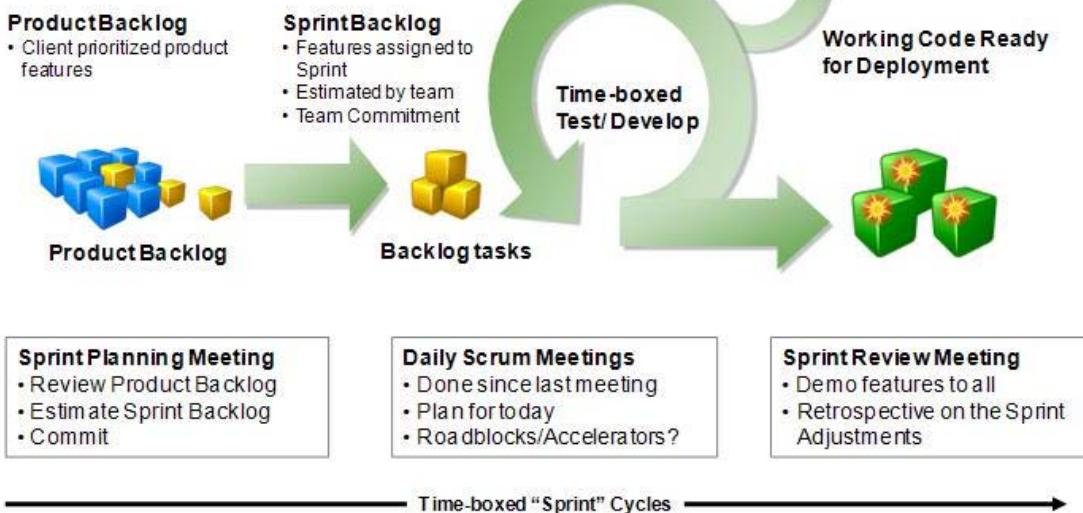
59



agile way of doing projects

- do the project in iterations (Scrum Sprints)
- after every iteration show demo to customer, and get comments/feedback
- if you do "agile" the wrong way, you may not succeed at the project
- agile way suits best for experienced professionals, and small projects
-

Scrum basics



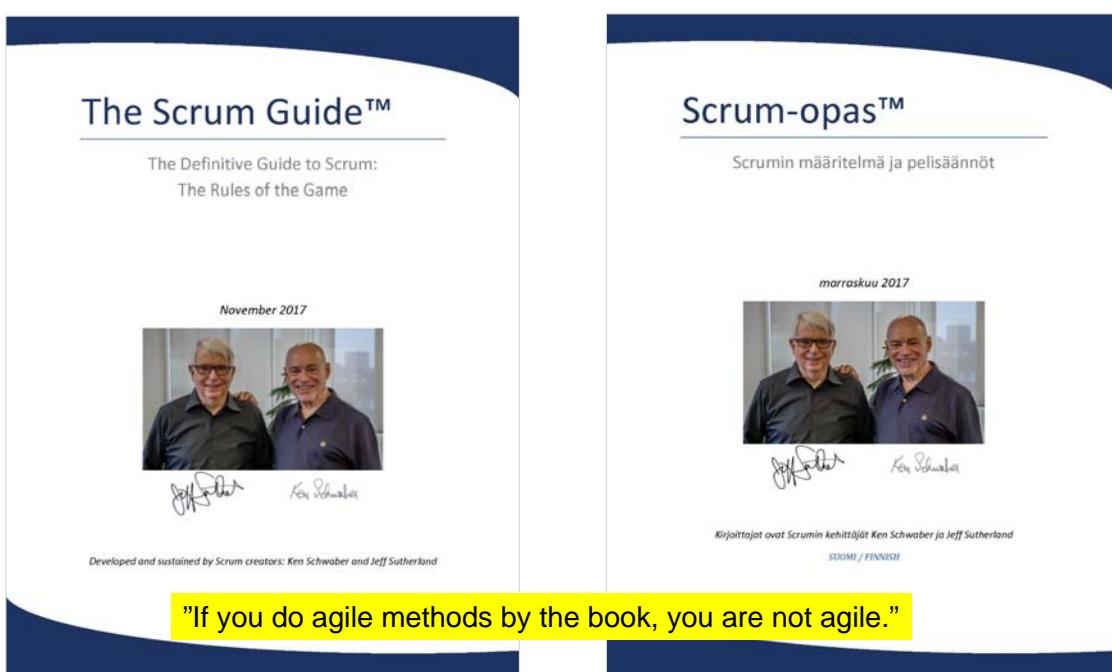
From Product Backlog (PB) you take a few features to Sprint Backlog (SB), and split those to tasks (BT).

03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

61

Get your own copy: www.scrumguides.org



03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

62

6point6 commissioned a survey of 300 CIOs in the UK and the US to examine their experiences of agile and measure how successfully the principles of agile are being applied and executed.

Chris Porter, CTO and co-founder of 6point6 and one of the authors of the report said: "Agile IT in the UK is facing a hidden crisis – **12% of agile projects are failing completely**".

You may fail, if you use agile "wrong".

Information Age

Diversity Events Newsletter Whitepa

News Data & Insight Sectors Topics The City & Wall Stre

Topics
IT management

UK wasting £37 billion a year on failed agile IT projects

British business is set to waste an estimated £37 billion on failed agile IT projects over the course of the next 12 months, according to a new report from independent IT consultancy 6point6

Nick Ismail
5 May 2017

f t e

68% of CIOs agree that agile teams require more architects. From defining strategy, to

[<https://www.information-age.com/uk-wasting-37-billion-year-failed-agile-it-projects-123466089/>]

TUNI * TIE-02306 Introduction to Sw Eng

03.12.2019

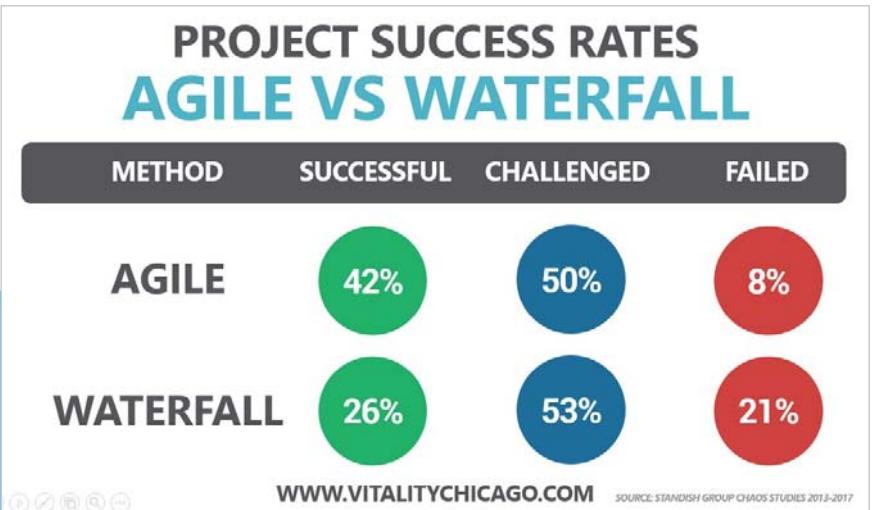
63

Standish group [via www.opendoorerp.com]

The top five factors found in successful projects are:

- User involvement
- Executive management support
- Clear Statement of Requirements
- Proper planning
- Realistic expectations.

There is still much room for improvements; less than half of agile projects were "successful".



TUNI * TIE-02306 Introduction to Sw Eng

03.12.2019

64

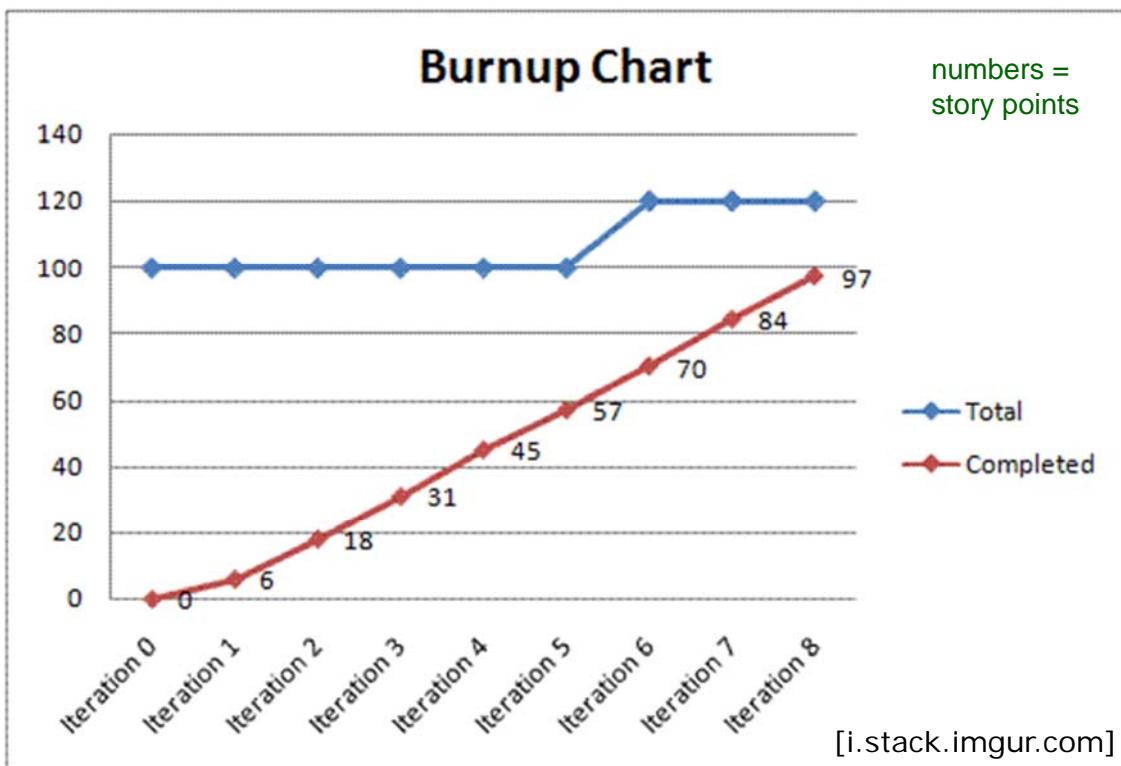
Success factors were:

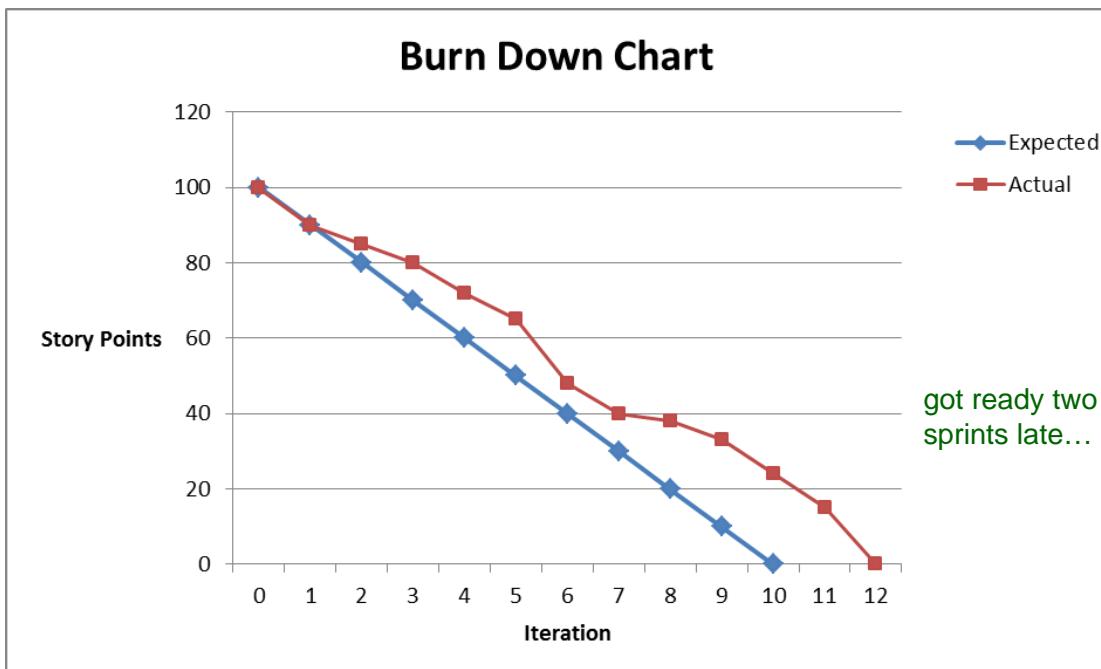
1. Good User Participation/ Involvement.
2. Good Requirements and Specifications.
3. Good of Skills.
4. Complete Requirements.
5. Good Management and Performance from Vendor/ Contractor
6. Good of Project Planning.
7. Realistic Expectations.
8. Good of Resources.
9. Good of Executive Support (Top Management).

Table 3. Frequency Of Failure Factors Selected By Respondents

Failure factors	Frequency	Percentage	Ranking
Lack of User Involvement	38	79.17	1
Changing Requirements and Specifications	36	75.00	2
Lack of Skills	35	72.92	3
Incomplete Requirements	34	70.32	4
Vendor/ Contractor	32	66.67	5
Performance and Management			
Lack of Planning	30	62.50	6
Unrealistic Expectations	29	60.42	7
Lack of Resources	27	56.25	8
Lack of Executive Support (Top Management)	24	50.00	9
Project Team Turn Over	18	37.50	10
Lack of Communication among project team	18	37.50	11
Project Complexity	18	37.50	12
Lack of User Knowledge	17	35.42	13
Lack of IT Management	16	33.33	14
Technology Illiteracy	15	31.25	15
Inter Department Communication	15	31.25	16

[Shamsudin Md Sarif et. Al.: Investigation of Success and Failure Factors in IT Project Management, 2018]





[blog.caplin.com]

03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

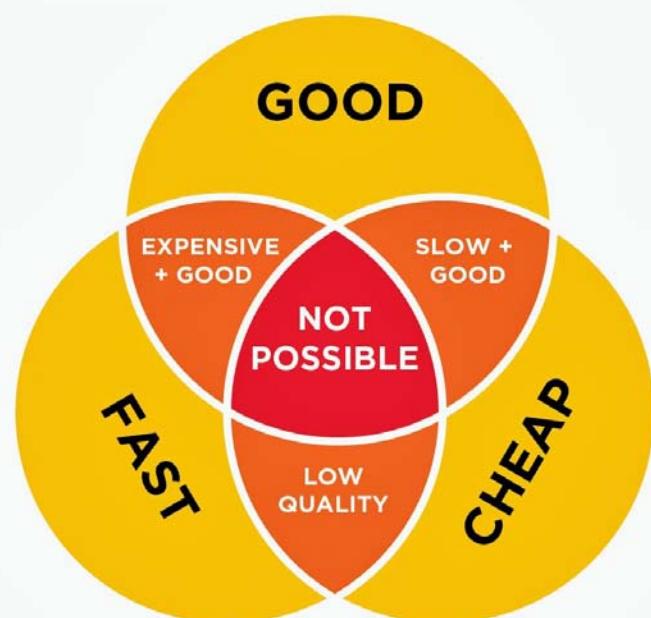
67



"How do you want your software ?
FAST - CHEAP – GOOD, pick two."



FOR CLIENTS WHO WANT IT ALL



TUNI * TIE-02306 Introduction to Sw Eng

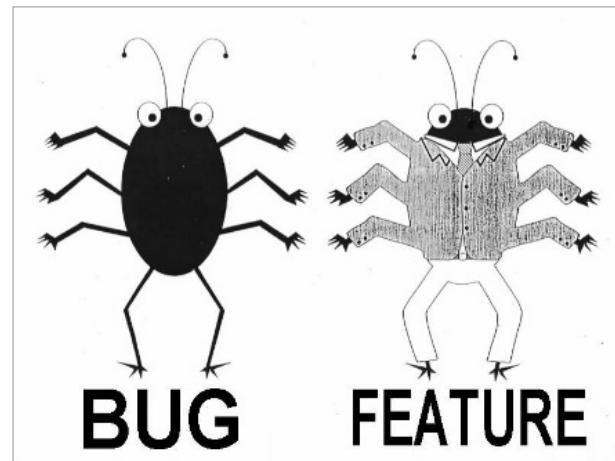
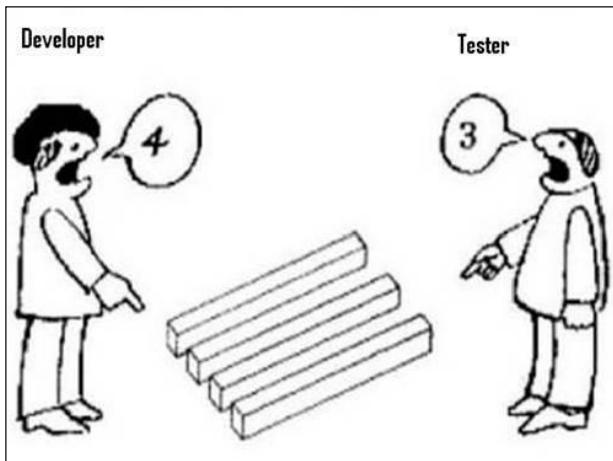
[hackernoon.com]

03.12.2019

68

Acceptance testing

- did customer got what was ordered ?
- customer role is important (e.g. right test data, real hardware environment, complete test cases, real end-users).



03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

69

<http://www.it-ehdot.fi/tutustu-ehtoihin>

IT2018-Sopimusehtoliitteet (pdf), esikatseluversiot

IT2018 YSE - yleiset sopimusehdot
IT2018 EAP - Erityisehtoja konsultointi- ja muista asiantuntijapalveluista
IT2018 EHK - Erityisehtoja henkilötietojen käsittelystä
IT2018 EJT - Erityisehtoja tietojärjestelmien ja asiakaskohtaisten ohjelmistojen toimituksista
IT2018 EKT - Erityisehtoja ohjelmistojen toimituksista ketterillä menetelmillä
IT2018 ELH - Erityisehtoja laitteiden huoltopalveluista
IT2018 ELT - Erityisehtoja laitetoimituksista
IT2018 EOY - Erityisehtoja ohjelmistojen ylläpitopalveluista
IT2018 ETP - Erityisehtoja tietoverkon välttyksellä toimitettavista palveluista (pilvipalvelu)
IT2018 EVT - Erityisehtoja valmisohjelmistojen toimituksista

Ladattavat IT2018-sopimusmallit (docx)

Henkilötietojen käsittelysopimus IT2018
Ketterien menetelmien toimitussopimus IT2018
Konsultointi- ja muita asiantuntijapalveluita koskeva sopimus IT2018
Laitteiden huoltopalvelusopimus IT2018
Ohjelmistojen ylläpitosopimus IT2018
Palvelutasokuvaus palvelun käytettävyyden mittauksesta IT2018
Salassapitosopimus IT2018
Tietoverkon välttyksellä toimitettavia palveluja koskeva sopimus IT2018
Toimitussopimus IT2018

IT 2018

Englanninkieliset IT2018-sopimusehtoliitteet (pdf), esikatseluversiot

IT2018 YSE - General terms and conditions
IT2018 EAP - Special terms and conditions for consulting and other professional services
IT2018 EHK - Special terms and conditions for the processing of personal data
IT2018 EJT - Special terms and conditions for deliveries of data systems and customised software
IT2018 EKT - Special terms and conditions for deliveries of software using agile methods
IT2018 ELH - Special terms and conditions for equipment maintenance
IT2018 ELT - Special terms and conditions for deliveries of equipment
IT2018 EOY - Special terms and conditions for software maintenance
IT2018 ETP - Special terms and conditions for services delivered via data network (cloud service)
IT2018 EVT - Special terms and conditions for deliveries of standard software

Ladattavat englanninkieliset IT2018-sopimusmallit (docx)

Agreement for the processing of personal data IT2018
Delivery agreement for software using agile methods IT2018
Consulting and other professional services agreement IT2018
Equipment maintenance agreement IT2018
Software maintenance agreement IT2018
Service level description on measuring usability of the service IT2018
Non-disclosure agreement IT2018
Agreement on services delivered via data network IT2018
Delivery agreement IT2018

03.12.2019

TUNI * TIE-02306 Introduction to Sw Eng

70

there are never too many testers

If you want to get reputation and respect, but you are not a skilled programmer, join some Open Source project (e.g. at GitHub) as a tester.

A good bug report requires "logical technical thinking".

If you feel cyber matters are what you do well, join some Bug Bounty programs to develop your skills and perhaps earn some money.

As a hobby (an advantage when applying for a job), it is recommended to know linux or some other *nix, and make own sw/hw projects.

Sw Dev

2019: How Many Software Developers Are There in the World?

According to Evans Data Corporation, there were **23 million** software developers in 2018, this number is expected to reach 26,4 million by the end of 2019 and 27,7 million by 2023.

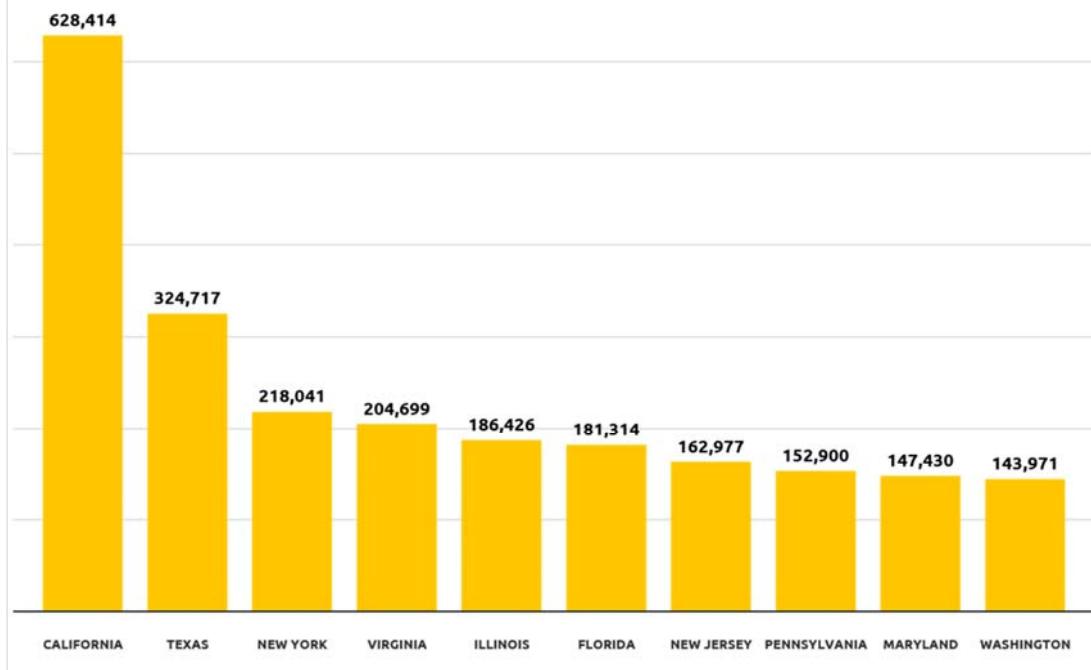
According to IDC calculations, in 2018 the number of software developers in the world grew to **22,3 million**, while in 2014 there were only 18,5 million programmers.

[<https://www.daxx.com/blog/development-trends/number-software-developers-world>]

So you think there is somewhere a job for you...?

TOP 10 US STATES WITH THE LARGEST NUMBER OF SOFTWARE ENGINEERS

DAXX



Highlights - What to remember

- customer's role is VERY important in sw projects
- agile projects fail if you do those the "wrong" way
- not all agile projects succeed (agile is not a "silver bullet")
- teamwork is essential in sw dev projects

- be careful about (non-critical) updates, sometimes updates may break the system
- If sw the tools are changing "every year", how you can get profitable with them ?

